DOCUMENT RESUME

ED 444 360 FL 026 350

AUTHOR Rogers, Paul
TITLE The Use of Commercial Application and Programming Software To Provide Language Students with Self-Access Materials Used in Their Own Time.
PUB DATE 1999-06-00
NOTE 21p.; Title also appears: Writing C.A.L.L. Self-Access Materials.
PUB TYPE Guides - Classroom - Teacher (052)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Classroom Techniques; *Computer Assisted Instruction; *Computer Software; *Computer Uses in Education; English (Second Language); *Independent Study; Internet; *Learner Controlled Instruction; Second Language Instruction; Second Language Learning; Teaching Methods

ABSTRACT

This paper describes the use of commercial application and programming software to provide language students with self-access materials used in their own time. In all cases, it assumes such students will be unsupervised. The idea is that in place of buying expensive and restricted commercially produced authoring packages, a teacher with limited computer skills can make his or her own tailor-made materials. The article begins with tips on using Microsoft PowerPoint to produce simple yet effective listening comprehension packages and goes on to suggest ways of using Visual Basic for reading, listening, and writing comprehension. The paper concludes with some simple comprehension work requiring students to download their own information from the Internet. The point is made throughout that it is especially important for the teacher to know precisely what he or she wants to do with the software package. Intensive and deliberate teacher planning and preparation are essential for computer uses in education to be effective and productive. Screen captures appear as needed throughout the text to make matters as clear as possible for the instructor. (Contains 9 Web sites and e-mail addresses where whole shell programs may be downloaded at no charge.) (KFT)

Reproductions supplied by EDRS are the best that can be made
from the original document.

ED 444 360

COVER PAGE

TITLE OF PAPER

The use of commercial application and programming software to provide language students with self-access materials used in their own time.

AUTHOR:

Dr. Paul Rogers

AFFILIATION:

Higher Colleges of Technology, United Arab Emirates

ACCEPTED BY:

The TEFL Farm

DATE:

June 1999

BEST COPY AVAILABLE

PERMISSION TO REPRODUCE AND DISSEMINATE THIS MATERIAL HAS BEEN GRANTED BY

Rogers, Paul

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☑ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

FL 026 350

Abstract

This paper describes the use of commercial application and programming software to provide language students with self-access materials used in their own time. In all cases, it assumes such students will be unsupervised. The idea is that in place of buying expensive and restricted commercially-produced authoring packages, a teacher with a smattering of computer skills can make his/her own tailor-made materials.

The article starts with using Microsoft PowerPoint to produce simple but effective listening comprehension packages and goes on to suggest ways of using Visual Basic for reading, listening and writing comprehension.

It concludes with some simple comprehension work requiring students to download their own information from the Internet. Some tentative conclusions and ideas for further study are then made.

Introduction

If we assume that most language learning environments today allow their students individual access to computers on a fairly routine basis, it makes good sense to provide them with material which is tailor-made to their specific course requirements. It has always seemed to me to be counter-productive to rely solely on commercially-produced language-learning software (e.g. ELLIS) unless an entire course is built around it.

> ***The analysis of software use can be applied to improve the design of new software applications. The orientation adopted is to analyze constraints of use through direct observations in real work situations***
>
> *Galer, M., Harker, S. & Ziegler, J p43*

A computer should act as a servant of the language teacher, never the master, yet it often appears that the professional producers would have us marvel at their bells, whistles and assorted multimedia, often at the expense of specific learning. Despite huge advances over the past few years, the use of computers in the language learning environment is still surprisingly limited:

> ***Man-computer communication is still predominantly monomedial: computer input is generated by manual means, mostly the keyboard, and output is presented visually, usually on a CRT. To enrich this communication speech may be applied, both as an input and output medium.***
>
> *Ob. cit. p280*

But until such time as we can routinely depend on realistic voice recognition and interactivity between *any* student and *any* machine, we should be content with the concept of computers as teachers' aids only. In this context, I contend they are uniquely suited to serve both us and our students and what follows is an attempt to utilize some of their power, ease and pleasure of use.

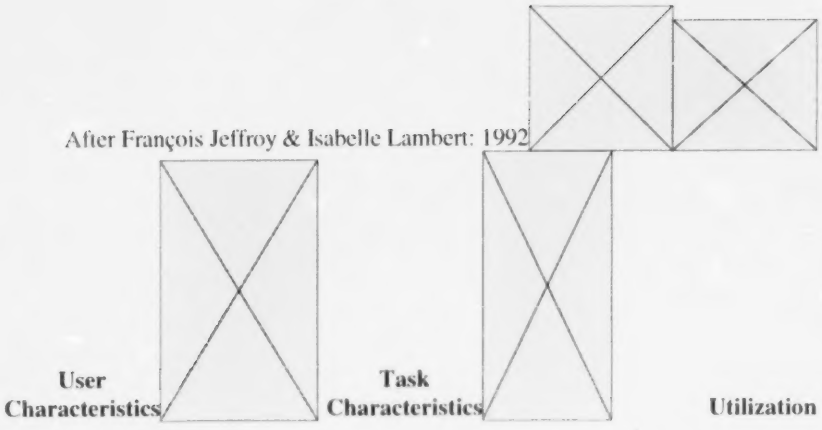
BEST COPY AVAILABLE

One Machine or Many?

Language schools will differ as to hardware and its availability for student use. I have known institutions which demanded computer literacy, and consequently installed hundreds of workstations in laboratory conditions in order to achieve it. The Higher Colleges of Technology in the United Arab Emirates, where I currently work, use networked machines as part of an Independent Learning Centre. Faculty are able to schedule their classes for specific periods in the ILC in order to encourage independent learning, while being on hand in case of insurmountable difficulty. At the lowest end of the scale is the small, private language school owning just a few machines, with inevitably restricted availability for students. While much of what follows assumes there are sufficient machines for each student in a class, there is no reason why several of the suggestions, if not all, can also be applied to the less well equipped establishments.

Methodology

Self-access should mean that the student is working independently, possibly unsupervised and yet is comfortable with the material s/he's being offered. In most cases, this limits the use of new and unfamiliar material. For the most part, therefore, self-access work should be in the area of consolidation and practice. It can also be useful for remedial or revision periods and if there is little or no supervision, these restrictions make multiple choice a natural option. We can apply the following model to consider the design constraints placed upon a writer of self-access programs:


After François Jeffroy & Isabelle Lambert: 1992
User Characteristics
Task Characteristics
Utilization
Future Software

I will have more to say about testing writing skills a little later, but for now, the chosen skill areas and methodology will be reading and listening practice via multiple choice. I am assuming familiarity with Windows 95 and specifically with PowerPoint (Office 97 version) while some of the later m/c material requires at least a passing acquaintance with Microsoft Visual Basic.

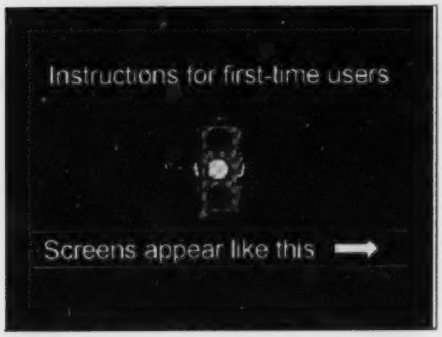
Listening Comprehension using *Microsoft PowerPoint*

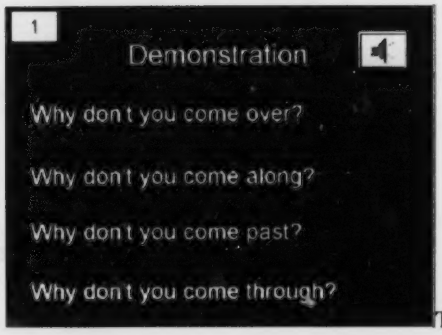
PowerPoint is conventionally used as a presentation application, in preference to an OHP. It can also be programmed to run autonomously as a stand-alone display; in an exhibition, for example. To do the latter, a given presentation is saved as a *.pps* extension, rather than *.ppt* (PowerPoint Show, rather than PowerPoint Presentation). It is the pps version I shall be concentrating on here.

In the Windows 95 environment, a PowerPoint Show can be iconized as a shortcut on the desktop, enabling students to access it directly with a mouse click. If the presentation has been saved as a PowerPoint Presentation, a shortcut simply gives you the working, or design version, and it will then need to be run as a "slide-show" by the student. Since we are working towards self-access, the pps extension gives the student immediate control of the given exercise. The application described below requires each student to have access to his/her own computer and to hear the relevant sound files on a headset.

Changes in conventional PowerPoint slides can be organized as mouse clicks or as timed automatic changes. For first-time users, I have designed some initial automatic

slides that give instructions using PowerPoint's own animations to make them clear.

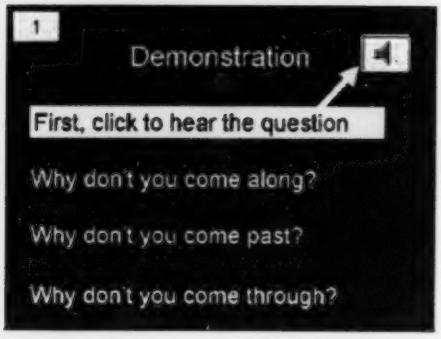




The pictures below are screen captures of

BEST COPY AVAILABLE

these instruction slides:



BEST COPY AVAILABLE

The student now knows that control of the exercise is in his/her hands. S/he can repeat a question, go forwards, quit, or start again. S/he is not being timed and no record of his/her "score" is being kept. This application is for review and practice only and there is no one around to mark it or make unfavourable comments (peers or teachers).

The big loudspeaker icon in the top right of the screen allows the student to hear the question. The possible answers then appear and s/he simply clicks the appropriate answer box. If the question has been answered correctly, the student hears cheering and applause. If the question has been answered wrongly, there is a sound of breaking glass. The student can repeat the choice of answers as often as s/he wants and a forward arrow at the bottom right hand corner of the screen is used only when s/he is ready to do so.

These different sound sources are accessed by Action Buttons from the Slide Show Menu. Clicking on this drop down list, you highlight Action Buttons and choose an appropriate icon. In the case of hearing the question, it made sense to use an internationally recognized symbol - a loudspeaker. Having chosen this, PowerPoint then asks the designer whether s/he wants a hyperlink to an external source (web, network, other program etc.) or simply a link to a specific file. In the case of this program, I pre-recorded all my listening questions as sound files appropriate to a given unit of the course I was writing (e.g. unit101.wav) and stored them in a special directory on my hard drive. Having employed the action button, it was then a simple matter, when asked for a link, to type in "C:\Sounds\unit101.wav" for the first

BEST COPY AVAILABLE

question, unit102.wav for the second and so on.

As far as the sound effects for right and wrong answers were concerned, I used Microsoft default sound files (applause and breaking glass respectively) which were accessed by pressing one of the action buttons (neutral yellow squares, chosen from the menu) to the right of each question.

I'd like to consider now the recording of the individual sound files. Since there is only one voice being used, with no need for elaborate sound effects - the intention is, after all, to test students' listening comprehension of spoken English - these questions can be recorded in mono which makes them easy to transfer to floppy disk if required. Within each PowerPoint question screen, the loudspeaker icon is used to capture the appropriate sound file question associated with the four multi-choice answers.

It therefore follows that the writing of the multi-choice format listening comprehension needs to be performed first. Once the questions, the correct answers and the three distracters have been written, the questions themselves can be recorded. The PowerPoint show is the last part of the writing.

Tests with students from a variety of first language backgrounds suggest that twenty-five questions is an optimal figure. More than this and students tend to become bored with the format. If fewer questions are chosen, the speed of a student's progress through the program may be too quick for him/her to benefit much.

Ideally, in reviewing English via self-access, computer assisted language learning should be used to test *both* receptive skills, not just listening. One way of combining the two is to use a programming language such as Microsoft Visual Basic.

Reading/Listening Comprehension using *Visual Basic*

The idea behind this project was to test students in a paperless examination context. In addition, the environment where it was first envisaged suffered notoriously from cheating. The brief, therefore, was to construct a testing system that was both paperless and "cheatproof." My subsequent use of it has again been in a self-access context since the basic program can be run by a student or students with no assistance from the teacher.

Questions were first written in a database (in this case, Microsoft Access). Visual Basic allows the use of various databases through "Dbase Manager". Six fields were employed: (1) Question, (2) Correct Answer, (3) Distracter 1, (4) Distracter 2, (5) Distracter 3 and (6) Question Number. Again, the total number of questions is up to the compiler. In the examination conditions under which this system was first run, there were 100: 50 listening, 50 reading.

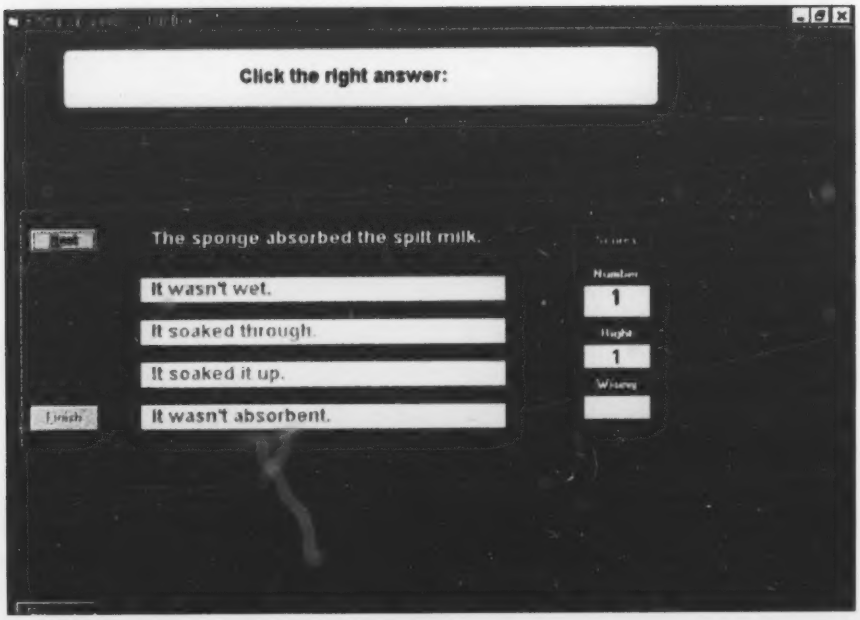
As with the PowerPoint exercises above, sound files can be attached to the database in place of a typed question field, so it is quite possible to mix reading with listening questions. The advantage of using a database question bank within Visual Basic is that the program allows *randomization*. In the example quoted below, not only are the question numbers randomized, the order of right answers and distracters is

BEST COPY AVAILABLE

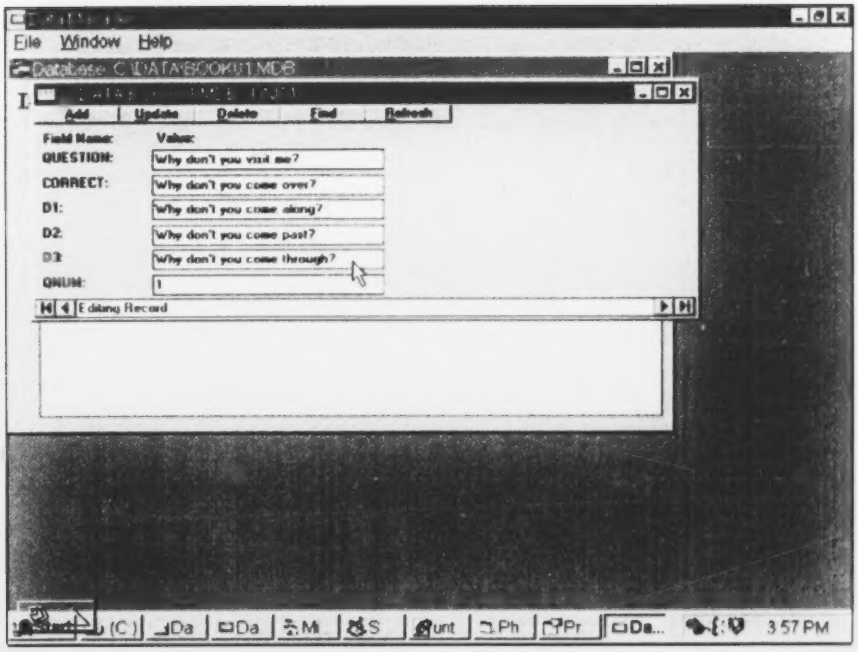
randomized *within* each question. This effectively prevents cheating in that Student A's question 1 may be Student B's question 6, and while Student A's right answer may be c), Student B's may be d).

There are many excellent texts and CD-ROMS currently available for those who have never used Visual Basic before. Anyone interested in the actual nuts and bolts of this program can have the code if you'd like to e-mail me with a request. I don't pretend that it's either very elegant or sophisticated but it does work.

Here are a couple of screen captures showing a simple Visual Basic design for multi choice and a sample of the Dbase fields from Microsoft Access:



BEST COPY AVAILABLE



BEST COPY AVAILABLE

Once a Visual Basic application is running to the programmer's satisfaction, it is a simple matter to make an *exec* file (from the main file menu). This, as with PowerPoint, can then be iconized as a shortcut on the desktop for immediate student access.

Self-Access Writing Exercises

I have also used Visual Basic as a programming platform for self-access writing exercises, but before describing this, mention must be made of the restrictions imposed by most computer software on creative writing.

Unless you are looking for a specific word or short phrase, I suggest that getting a computer to assess creative written expression is fraught with difficulty for the average amateur programmer. Simply, the sheer range of possible options of syntax, lexis, punctuation and style in longer pieces of writing preclude most uses of computer assessment. However, there is no reason why a student can't be working on his/her own at a workstation, producing text which can be accessed by a teacher for marking and assessment later.

In similar vein to the multi choice program mentioned above, I designed a Visual Basic interface that accesses questions from a Dbase, that only contains two fields - the question and the question number. These questions are very open ended, allowing considerably more student input than the mouse clicks of multiple choice. These responses are recorded in a student's own text file which has already been named by the first question ("First, please enter your name"). A file now exists for all that individual student's answers.

On completion of the exercises, the next Visual Basic screen prints the student's answers in full, alongside all the questions. The text box containing the answers

enables corrections or revisions and only when the quit button is pressed will the final version be saved under the student's name.

The teacher then has two options. S/he can either access the file direct at a later time and correct it on screen (perhaps in a different colour text), or, which is what I prefer to do, print off hard copies of the answers, mark them in the conventional way and then hand them back to individual students concerned.

I enclosed a sub-routine in the basic program which overwrites previous files automatically, otherwise an individual student file would grow inordinately. The drawback with this is that a teacher must access and mark a student file before that student next uses the machine. This has the slight 'advantage' of imposing a marking discipline on the teacher.

Using the Internet for Self-Access Comprehension

In the summer of 1998, I was fortunate enough to teach a short ESP course in Aviation English for Arab military personnel. As an adjunct to this course, I also had to teach some basic computer literacy and it seemed an excellent idea to marry the two courses by giving the students task-based work via the Internet. The first step was to help each student acquire his own e-mail address which we did using MS Hotmail. We then practised sending mail to each other and within the group. Finally, I posted assignments to each student's address each morning and their independent learning task for the day was to check their mail, access the task, find the relevant answers to the questions set and then e-mail their results back to me. In this way, individual tasks could be set for individual students. This can be time-consuming for the teacher of course, but with a small group (six) and a light teaching load, I was able to "mix and match" so that everyone had his own assignment for the day. In addition, once the task had been written, it can be used over and over again, provided one checks the

relevant web pages regularly to ensure the information used is still current.

Initially, the students were very limited and we had to proceed slowly. However, by the end of the first week, they were confident enough with Netscape to start playing around with Search Engines. From then on, the tasks became increasingly ambitious. The following is a typical student's daily task:

Your subject for today is Boeing, the American aircraft maker.
Follow the instructions below.

1. In "Location" type the following address: **http://www.boeing.com**
2. After a few seconds the Boeing homepage will appear on the screen. On this page click on the words "Military planes".
3. After a few seconds a new page will appear on your screen. From this page choose "Fighters and Bombers."
4. Select the drop-down menu and click on "F-15 Eagle."

Now see if you can find the answers to these questions. Remember, they may not all be found on the same page!

Questions

1. When did the F-15 enter service?
2. How long will Boeing continue to make the F-15?
3. Who makes the F100-229 engines that power the F-15?
4. What do the letters "LANTIRN" mean?
5. Who makes the F-15 in Japan?
6. Approximately how many F-15s are in service worldwide?
7. How long, high and heavy is an F-15?
8. What is its top speed?
9. What are the popular names for AIM-9 and AIM-7 missiles?
10. What did Boeing first demonstrate on August 6th, 1998?

BEST COPY AVAILABLE

Conclusion

> ***All conventional techniques are based on the paradigm that the user carries on a sort of dialogue or interacts with the machine which is clearly a technical entity only accessible through specific devices like keyboards, buttons and the like. In virtual realities, the coupling between user and computer becomes much closer. Headmounted 3D displays and body sensors like data gloves, data suites or exoskeletons introduce the user into a virtual world in which he can move, perceive and act in close analogy to the real physical environment. .... Users may explore buildings before they are built, remote operations in dangerous environments become possible or students may analyze complicated chemical or biological structures in sophisticated learning environments.***
>
> Galer, M., Harker, S. & Ziegler, J. *p420*

We can assume that with ever-sophisticated technology, computers in the hands of a language teacher will become ever-easier to control and with which s/he can create exciting, relevant and pedagogically sound materials. Most recently, I have been working with Assymetrix Toolbook II to create self access language programs that allow use of sound, video, true/false, drag/drop and all kinds of multi-choice or text-based material. However, there is a special danger here as far as the average language teacher is concerned: regardless of how easy the package may be to use, do you, the teacher, know *precisely* what you want of it?

Some years ago, I was teaching a group of young language teachers how to use video in the classroom and as a rule of thumb suggested that for every five hours of preparation, they might get five *minutes* of usable original video material. Consider this: when we buy a video camera (or a computer) we are not acquiring the craft skills alongside the equipment - these have to be learned like any other. Now, although computer programs make life increasingly easy for the novice to use, there is little if any, guidance as to how to organize your thoughts and ideas beforehand.

In other words, think of how a professional company organizes its language learning software. There will be a team: probably designer, programmer, language advisor,

marketer and so on. If you want to design your own applications, you, the full-time teacher will have to wear all these hats and still produce something that looks good, works well and is genuinely useful for your students. It's a tall order.

There is only one way to ensure you succeed: plan carefully beforehand exactly what you want your program to do and how you want it to work. Ask yourself realistically, "Is there nothing on the software market that can do this job for me?" And having decided to go it alone, there's absolutely no point in thinking "I'll just do multiple choice." You have to ask yourself all sorts of questions. For example, do you want your students to have immediate or delayed feedback to their answers? Do you want to incorporate listening into your questions or will they only be text based? Do you want the questions randomized, and if so, do you want the same to apply to the answers? Are you going to employ colours, graphics, animations and so on?

The only successful way to work is to storyboard your application beforehand. This means sitting down with paper and pencil and drawing each and every screen, layout, question/answer format, bells and whistles you want to use.

In the bibliography that follows, I have included websites and e-mail addresses where you can download whole shell programs at no charge and just type in your questions. This may be all you need. However, if you want to try your hand with some of the ideas above, please feel free to contact me directly. My e-mail address is added to the biographical details that end this paper. I'm always glad to hear from people and will help all I can. Just be sure beforehand that you know what you want your program to do and what its limitations are. Think it all through as carefully as you can and you won't go far wrong.

Bibliography

Winograd, T. & Flores, F. *Understanding computers and cognition*, Ablex Pub. Corp., Norwood, New Jersey, 1986

Carroll, J.M. *Feeding the interface eaters*, ***People & Computers V***, Cambridge University Press, 1989

Stahl, _. *Testing for usability can head off disaster*, ***Computer World***, December 1987

Hodges, M.E., Sasnett, R.M. & Ackerman, M.S. *A construction set for multimedia applications* ***IEEE Software***, January, 1989

Galer, M., Harker, S. & Ziegler, J. *Methods and Tools in User-centred Design for Information Technology* North-Holland, Amsterdam, 1992

Internet websites and e-mail addresses

http://www.les.aston.ac.uk
Homepage of Language Studies Unit, Aston University, Birmingham, UK

http://www.net-shopper.co.uk/creative/education/languages/martin
Free programs from Martin Holmes - Choices and Cloze are easily downloaded shell programs for multiple choice and Cloze-writing. Web-based free software is also very impressive.

http://www.tes.co.uk
Homepage for Britain's Times Educational Supplement

http://www.edunet.com
Digital Education Network

http://www.toefl.org
US-based exam specialists

http://www.ucles.org.uk
UK-based exam specialists

http://www.mlc-dl.com
My own homepage

bradsaun@emirates.net.ae
One of the most experienced and able computer-TESOL teachers in the Middle East

BEST COPY AVAILABLE

ian.pople@man.ac.uk
Applied Linguist at Manchester University, UK - excellent for academic TESOL

Biographical details

Paul Rogers has been teaching English to foreign students in many areas of the world for the past twenty-five years. He is currently based in Abu Dhabi, as General Education Curriculum Coordinator for the ten Higher Colleges of Technology in the United Arab Emirates. He is an external tutor for Aston University, Birmingham, UK, for whom he has run several Teacher Training programs, the latest of which is Distance Learning and web-based.

He has worked extensively in curriculum development and teacher training and has considerable computer experience both as a language teacher and teacher trainer.

He is married to Helen, a Greek Cypriot, herself an experienced EFL teacher and they have four children and one granddaughter.

Paul welcomes any inquires about the current paper, or on any subject related to language teaching, and can be reached at:

Academic Services
Higher Colleges of Technology
PO Box 32092
Abu Dhabi
UAE

or on e-mail at:

paul.rogers@hct.ac.ae